

What is claimed is:

1 1. A method for network packet processing comprises:
2 receiving network packets; and
3 operating on the network packets with a plurality of program
4 threads to affect processing of the packets.

1 2. The method of claim 1 wherein operating comprises:
2 using at least one program thread to inspect a header
3 portion of the packet.

1 3. The method of claim 1 wherein operating further
2 comprises:
3 signaling by the at least one program thread that a packet
4 header has been processed.

1 4. The method of claim 1 wherein the plurality of program
2 threads are scheduler program threads to schedule task orders for
3 processing and processing program threads that process packets in
4 accordance with task assignments assigned by the scheduler
5 program threads.

1 5. The method of claim 1 wherein each program thread
2 writes a message to a register that indicates its current status.

1 6. The method of claim 5 wherein interpretation of the
2 message is fixed by a software convention determined between a
3 scheduler program thread and processing program threads called by
4 the scheduler program thread.

1 7. The method of claim 5 wherein status messages include
2 busy, not busy, not busy but waiting.

1 8. The method of claim 5 wherein a status message includes
2 not busy, but waiting and wherein the status of not busy, but
3 waiting signals that the current program thread has completed
4 processing of a portion of a packet and is expected to be
5 assigned to perform a subsequent task on the packet when data is
6 made available to continue processing of the program thread.

1 9. The method of claim 5 wherein the register is a
2 globally accessible register that can be read from or written to
3 by all current program threads.

1 10. The method of claim 4 wherein scheduler program threads
2 can schedule any one of a plurality of processing program threads
3 to handle processing of a task.

1 11. The method of claim 10 wherein the scheduler program
2 thread writes a register with an address corresponding to a
3 location of data for the plurality of processing program threads.

1 12. The method of claim 11 wherein a selected one of the
2 plurality of processing program threads that can handle the task
3 reads the register to obtain the location of the data.

1 13. The method of claim 12 wherein the selected one of the
2 plurality of processing program threads reads the register to
3 obtain the location of the data and to assign itself to
4 processing the task requested by the scheduler program thread.

1 14. The method of claim 12 wherein the selected one of the
2 plurality of processing tasks reads the register to obtain the
3 location of the data, while the register is cleared by reading
4 the register by the program thread to assign itself to process
5 the task.

1 15. The method of claim 13 wherein when another one of the
2 plurality of processing program threads assignable to the task
3 attempts to read the register after it has been cleared, it is

4 provided with a null value that indicates that there is no task
5 currently assignable to the processing program thread.

1 16. A parallel hardware-based multithreaded processor for
2 receiving network packets comprises:

3 a general purpose processor that coordinates system
4 functions; and

5 a plurality of microengines that support multiple program
6 threads, and operate on the network packets with a plurality of
7 program threads to affect processing of the packets.

1 17. The processor of claim 16 wherein one of the plurality
2 of microengines executes scheduler program threads and remaining
3 ones of the microengines execute processing program threads.

1 18. The processor of claim 16 further comprising a global
2 thread status register wherein each program thread writes a
3 message to the global status register that indicates its current
4 status.

1 19. The processor of claim 18 wherein interpretation of the
2 message is fixed by a software convention determined between a

3 scheduler program thread and processing program threads called by
4 the scheduler program thread.

1 20. The processor of claim 16 further comprising:

2 a read once register, wherein the scheduler program thread
3 writes the read once register with an address corresponding to a
4 location of data for the plurality of processing program threads
5 and when a selected one of the plurality of processing program
6 threads reads the register to obtain the location of the data,
7 assigns itself to processing the task requested by the scheduler
8 program thread, while the register is cleared by reading the
9 register by the program thread.

1 21. The processor of claim 20 wherein when another one of
2 the plurality of processing program threads assignable to the
3 task attempts to read the read once register after it has been
4 cleared, it is provided with a null value that indicates that
5 there is no task currently assignable to the processing program
6 thread.

1 22. An apparatus comprising a machine-readable storage
2 medium having executable instructions for network processing, the
3 instructions enabling the apparatus to:

4 receive network packets; and
5 operate on the network packets with a plurality of program
1 threads to affect processing of the packets.

1 23. The apparatus of claim 22 wherein instructions to
2 operate further comprise instructions to:
3 use at least one program thread to inspect a header portion
4 of the packet.

1 24. The apparatus of claim 22 further comprising
1 instructions to provide scheduler program threads to schedule
2 task orders for processing and processing program threads to
3 process packets in accordance with task assignments assigned by
4 the scheduler program threads.

1 25. The apparatus of claim 22 wherein each program thread
2 writes a message to a register that indicates its current status.

1 26. The apparatus of claim 25 wherein the register is a
2 globally accessible register that can be read from or written to
3 by all current program threads.

1 27. The apparatus of claim 22 wherein the scheduler program

2 thread writes a register with an address corresponding to a
3 location of data for the plurality of processing program threads
1 and a selected one of the plurality of processing program threads
2 that can handle the task reads the register to obtain the
1 location of the data, and clears the register after reading by
2 the program thread.

1 28. The apparatus of claim 27 wherein when another one of
2 the plurality of processing program threads assignable to the
3 task attempts to read the register after it has been cleared, it
4 is provided with a null value that indicates that there is no
5 task currently assignable to the processing program thread.